SANDIA REPORT

SAND2015-4087
Unlimited Release
Printed May 2015

Cyber Graph Queries for Geographically Distributed Data Centers

Jonathan Berry, Michael Collins, Aaron Kearns, Cynthia A. Phillips, Jared Saia

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2015-4087
Unlimited Release
Printed May 2015

Cyber Graph Queries for Geographically Distributed Data Centers

Jonathan Berry, Computing Research
Mail Stop 1327
P.O. Box 5800
Albuquerque, NM 87185

Michael Collins, Physics, Computer Science & Engineering
Christopher Newport University
1 Avenue of the Arts
Newport News, VA 23606

Aaron Kearns, Computer Science Department
University of New Mexico
Farris Engineering Building
Albuquerque, NM 87131-1386

Cynthia A. Phillips, Computing Research
Mail Stop 1326
P.O. Box 5800
Albuquerque, NM 87185

Jared Saia, Computer Science Department
University of New Mexico
Farris Engineering Building
Albuquerque, NM 87131-1386

Abstract

We present new algorithms for a distributed model for graph computations motivated by limited information sharing we first discussed in [20]. Two or more independent entities have collected large social graphs. They wish to compute the result of running graph algorithms on the entire set of relationships. Because the information is sensitive or economically valuable, they do not wish to simply combine the information in a single location. We consider two models for computing the solution to graph algorithms in this setting: 1) limited-sharing: the two entities can share only a polylogarithmic size subgraph; 2) low-trust: the entities must not reveal any information beyond the query answer, assuming they are all honest but curious. We believe this model captures realistic constraints on cooperating autonomous data centers.

We have algorithms in both setting for *s*-*t* connectivity in both models. We also give an algorithm in the low-communication model for finding a planted clique. This is an anomaly-detection problem, finding a subgraph that is larger and denser than expected. For both the low-communication algorithms, we exploit structural properties of social networks to prove performance bounds better than what is possible for general graphs. For *s*-*t* connectivity, we use known properties. For planted clique, we propose a new property: bounded number of triangles per node. This property is based upon evidence from the social science literature.

We found that classic examples of social networks do not have the bounded-triangles property. This is because many social networks contain elements that are non-human, such as accounts for a business, or other automated accounts. We describe some initial attempts to distinguish human nodes from automated nodes in social networks based only on topological properties.

Acknowledgment

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Contents

List of Figures

Chapter 1

Introduction

Consider two entities, Alice and Bob, who autonomously observe the world, collecting information on social relationships, which each represents as a social graph. Alice would like to combine her information with Bob's to answer a query about the full set of relationships. It is in Bob's best interest to cooperate, since he may need Alice's help in the future. But there are barriers to total information sharing, which we model in two ways: 1) limited-sharing: Alice and Bob can share only a polylogarithmic size subgraph; 2) low-trust: Alice and Bob must not reveal any information beyond the query answer, assuming they are both honest but curious.

We are motivated by recent trends in data collection over large social networks. Specifically, we consider a small number of autonomous data centers that are collecting data about a social network. Periodically, these centers may want to collaborate to solve a computational query. However, data is a critical resource, so the centers want to answer the query while sharing as little data as possible.[1]

Brickell and Shmatikov [3] were similarly motivated when they conducted related work using a different model. They provide several compelling examples involving commercial entities that must evaluate the consequences of a potential merger, or coordinate in some useful way without revealing private details. Networking companies would be interested in measuring the efficiency of joint infrastructure before committing to a merger, shipping companies would similarly need to know the effects of merged capacities on efficient routing, and social networking websites may wish to collaborate to compute more accurate statistical measures of their users' behavior without revealing private information.

Our cooperative computing problem has overlap with two mature research areas that deal with privacy: secure multiparty computation and differential privacy. In secure multiparty computation (MPC), a set of m parties, each of whom has a private input, want to compute an m-variate function over their inputs, without revealing any information about their inputs (see e.g. [18] for a survey of MPC). A novelty of our problem when compared to most results in MPC is that the size of the inputs held by the parties are very large. In differential privacy, a single entity holds all the data, and the goal is to answer queries as accurately as possible, while minimizing the chance of leaking information about individual records in that data (see e.g. [10]). By contrast, in our problem, multiple entities hold the data. The entities seek to minimize not the chance of identifying

[1] We imagine that user data may be valued more by data centers than it is valued by individuals, since the data centers can monetize that data.

individual records in the data, but rather the total amount of information revealed about their own data set. Also, they require that the query is answered exactly.

The term "data center" frequently refers to a distributed set of resources owned by a single entity such as Google, cloud systems, or providers of web services. Cooperation in such settings is a given, with research focusing on providing quality of service while minimizing energy or other costs. See for example these surveys [1, 17]. In our setting, the data centers are owned by autonomous, potentially competing parties who nevertheless wish to compute cooperatively in some cases while minimizing the loss of proprietary information.

1.0.1 Our Model

Our model assumes a small number of autonomous data centers. For simplicity of discussion here, we will assume two centers, but our results for *s-t* connectivity generalize to any small constant number of centers. Let G_a be Alice's graph and G_b be Bob's graph. Alice and Bob wish to perform computations on the graph $G_U = G_a \cup G_b$. Let n be the number of nodes in G_U. Alice and Bob build their graphs by observing a common world graph G. The fundamental observation is an edge representing a relationship between two people. Alice and Bob know nothing about each other's graphs. However, the nodes come from a shared namespace, so if Alice gives Bob an edge (or vice versa), he will recognize the nodes if he has seen them before, and therefore he knows where that edge fits into his graph.

The world graph G is a social network, and therefore has topological properties of a social network. In general, Alice and Bob can each sample from this graph according to arbitrary distributions. Thus theoretically, G_a, G_b, and G_U do not necessarily inherit social network topological properties in the worst case. However, every example of a social network that researchers have observed to determine the currently accepted set of properties of such networks is itself a sampling of the world graph of all human relationships. We assume Alice's and Bob's samples are gathering in ways that also produce these classic properties.

1.0.2 Results

In the *s-t* connectivity problem, we wish to find a path between two specified vertices s and t. Consider the graph in Figure 1.1(b). There is a path from vertex 37 to vertex 9. If that graph is distributed to three data centers as showing in Figure 1.1(a), then no one data center has enough information to determine that these two nodes are connected. That conclusion requires an edge from each of the data centers. Thus the data centers working together can compute a path that none can compute on its own.

We published results for computing *s-t* connectivity in both the low-communication model and in the low trust model in [2]. That paper includes a discussion of related models. Here is the relevant part of the abstract from [2]:

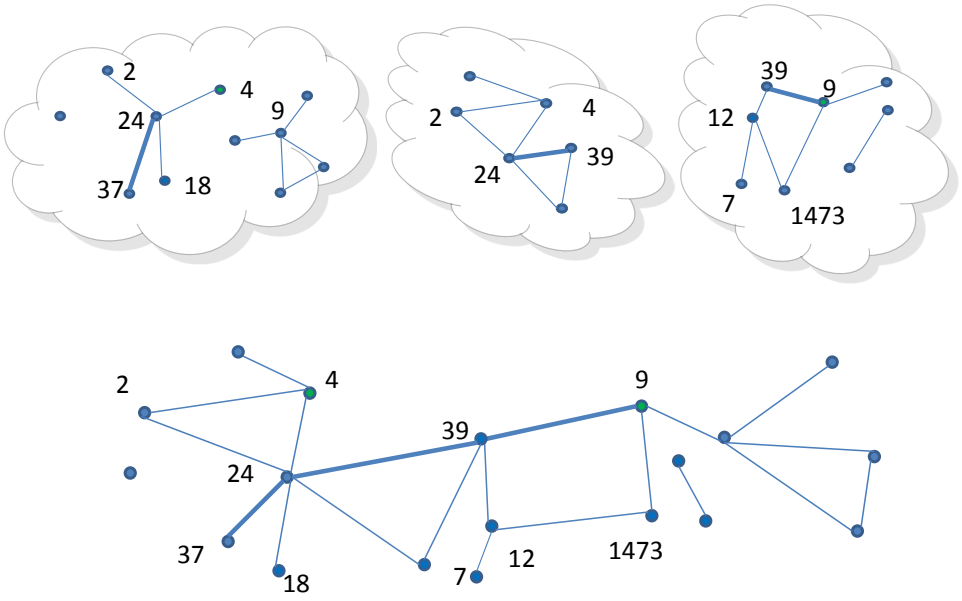

Figure 1.1: An example of a distributed graph (from [20]). The graph in on the bottom is distributed among three data centers shown at the top. Only by combining information from all three centers can one infer that nodes 37 and 9 are connected.

In the limited-sharing model, our results exploit social network structure. Standard communication complexity gives polynomial lower bounds on *s*-*t* connectivity for general graphs. However, if the graph for each data center has a giant component and these giant components intersect, then we can overcome this lower bound, computing *s*-*t* connectivity while exchanging $O(\log^2 n)$ bits for a constant number of data centers. We can also test the assumption that the giant components overlap using $O(\log^2 n)$ bits provided the (unknown) overlap is sufficiently large.

The second result is in the low trust model. We give a secure multi-party computation (MPC) algorithm that 1) does not make cryptographic assumptions when there are 3 or more entities; and 2) is efficient, especially when compared to the usual garbled circuit approach. The entities learn only the yes/no answer. No party learns anything about the others' graph, not even node names. This algorithm does not require any special graph structure. This secure MPC result for *s*-*t* connectivity is one of the first that involves a few parties computing on large inputs, instead of many parties computing on a few local values.

In Section 2 we give an efficient algorithm for finding a planted clique. A clique is a graph with all possible connections. The planted-clique problem is motivated by anomaly detection: finding subgraphs that are denser than expected. Finding the largest clique in a general graph is NP-hard and polynomially hard to approximate [11]. However, we exploit properties of social networks to find a planted clique that is larger than the largest clique a social network would natively have. Finding (large) planted cliques is made particularly tractable by our assumption,

defended in Section 2, that the number of nodes in the largest clique in a social network is bounded by a constant. Intuitively, this is motivated by human limitations on time and attention, and the need to cultivate strong relationships.

More specifically, we make two assumptions about the social network: 1) the maximum degree is $O(n^{1-\varepsilon})$, for $\varepsilon > 0$, a standard assumption for social networks [4, 6] and 2) the clustering coefficient of a node with degree d is $O(1/d^2)$. The second assumption is equivalent to having a bounded number of triangles per vertex, and it implies a constant-sized maximum clique. Our goal is to find the planted clique while minimizing communication between the players.

We give a protocol that provably ensures with high probability (whp) that both players find the clique, while requiring at most polylogarithmic communication, and polynomial computation. We believe our algorithm can be adapted to find a more generalized planted dense graph such as a γ-quasi-clique. A γ-quasi clique is a graph with at least a γ fraction of the maximum possible number of edges.

In Section 3 we describe why social network snapshots available on the web typically do not have one of the properties required for correctness of the planted clique algorithm. Social networks are a combination of a human sub-network and an automated subnetwork. Accounts run by machines are not subject to the same limitations humans are. We describe some initial work to extract the human subcomponent of a network. We are initially motivated by our need for real data to validate the algorithm from Section 2. However, there are other applications for methods to classify nodes in a network as human vs non-human. For example, spam sub-networks will generally consist of non-human nodes.

Chapter 2

Finding a Planted Clique

In this section, we describe, and prove/sketch the correctness of, an algorithm for finding a planted clique in a social network. Let n be the number of nodes in the social network. We assume that a subset of $O(\ln n)$ nodes, S, are chosen uniformly at random and that edges are added among these nodes to form a clique. The adversary distributes the edges arbitrarily to Alice and Bob subject to the constraint that at least one player knows each planted clique edge. Some edges of the base graph may be in neither graph, but this only makes the problem easier.

For a node v with degree d, the *clustering coefficient* of that node is

$$\frac{\text{Number of triangles containing node v}}{\binom{d}{2}}.$$

The denominator is the number of possible triangles on node v, one for each pair of neighbors. Thus the clustering coefficient for a node v is the fraction of possible triangles that node v participates in. A high clustering coefficient (close to 1) implies that node v's neighbors are strongly connected to each other.

For our planted clique algorithm, we make two assumptions about the social network: 1) the maximum degree is $O(n^{1-\varepsilon})$, for $\varepsilon > 0$, a standard assumption for social networks [4, 6] and 2) the clustering coefficient of a node with degree d is $O(1/d^2)$. The second assumption implies a constant number of triangles per node.

We give an algorithm to find a planted clique of size $O(\log n)$ in an n-node graph distributed between two data centers using polylogarithmic communication. This appears to be much easier than finding cliques in half-dense non-distributed Erdös-Renyi graphs, where the largest clique is of size $O(\ln n)$, but the best algorithms can only find planted cliques of size $\Theta(\sqrt{n/e})$ [8].

2.0.3 A new social network property

We first justify the second assumption. Sociologists have argued that the number of strong links that a node can have in a social network is bounded, even for online social networks [9]. That is, the increased power of computers and the internet cannot overcome basic human limitations on time and attention paid to another person. Other than immediate family ties, which are bounded, strong links between people generally require consistent effort over time.

Easley and Kleinberg [7] argue the *triadic closure property*: if node v has a strong link to node u and node v has a strong link to node w, then nodes u and w are more likely to be connected, at least weakly, than a random pair of nodes in a network. Easley and Kleinberg give three reasons. The first is opportunity. Because node v knows, and presumably frequently interacts with, node u and w, (s)he has opportunities to introduce u and w to each other. The second reason is transitive trust. Nodes u and w both trust node v. Therefore, they are likely to pay more attention to each other when introduced by a mutual trusted friend. The third reason is social stress. This applies more to some groups, such as teenage girls, than to others. If node w is spending time with node u, then she is not spending time with node w and vice versa. This causes stress on both relationships. It is frequently less stressful for node v to do things with both u and w than to exclude one of them.

We posit that the converse of the triadic closure property is also true. That is, we posit that if one or more of the links (v,u) and (v,w) are weak, then there is no (or significantly reduced) increased probability that node u and w will become connected through their relationship with node v. All three reasons for node v to introduce nodes u and w decrease with decreased strength of ties. Kossinets and Watts [14] corroborate this with an experiment involving students. Their data shows that the probability of a new relationship mediated by a mutual friend/acquaintance is directly proportional to the average strength of the ties to the third (mutual friend) node. Thus we assume that if a node has a constant amount of resources for such facilitation and most are devoted to strong links, then as degree increases, the probability of facilitating a relationship between non-strong pairs decreases quadratically. The assumption that the clustering coefficient of a degree-d node $O(1/d^2)$, is also a property implied with high probability from the per-degree clustering coefficient expression posited by Kolda et. al. [13].

2.1 Algorithm Sketch

Let G_a be Alice's graph and let G_b be Bob's graph. Alice and Bob run the following algorithm. They also run the algorithm with their roles reversed and return the largest of the two cliques. If any set is too large to send (super-polylogarithmic), just stop (the adversary gave Alice too few triangles from S).

The Planted-Clique-Finding Algorithm:

1. Alice finds the subset of nodes, called Q_a, with maximum triangle density. Triangle density is the number of triangles divided by the number of nodes. We can find Q_a in polynomial time using a triangle-density version of the edge-density linear program in [5].

2. Alice finds the set of nodes, $\bar{N}_a(Q_a)$, each adjacent to at least half the nodes in Q_a in the graph G_a. She sends Q_a and $\bar{N}_a(Q_a)$ to Bob.

3. Bob computes $\bar{N}_b(Q_a)$, the set of nodes each adjacent to at least half the nodes Q_a in G_b. Let $V_C = Q_a \cup \bar{N}_a(Q_a) \cup \bar{N}_b(Q_a)$. Bob finds the set of edges, E_b, induced by the nodes V_C in G_b. He sends E_b and $\bar{N}_b(Q_a)$ to Alice.

4. Alice finds E_a, the set of edges induced by V_c in G_a.

5. Alice finds the maximum clique in the graph $(V_C, E_a \cup E_b)$ using any algorithm guaranteed to find the maximum clique such as [19].

In step 1, in practice, one might like to use a faster approximation. In [5], Charikar gives a greedy 2-approximation for finding a maximum edge-density subgraph. The obviously generalization, using triangle counts of a node instead of degree, gives a three-approximation for finding a maximum-triangle-density subgraph. The proofs in [5] extend to triangles with no major changes.

2.2 Correctness Sketch

Let S be the nodes in the planted clique. We first show using Ramsey theory that one of Alice or Bob will receive $\Theta(\ln^3 n)$ triangles of S.

Lemma 1. *At least one player gets $C' log^3 n$ of the triangles in S for some constant C'.*

Proof. Consider the set of 6-cliques within the planted clique S. For notational convenience let x denote the number of nodes in S. Color the edges in S red or blue depending on whether they belong to the subgraph possessed by Alice or Bob respectively. If an edge belongs to both Alice and Bob, color it arbitrarily. A known Ramsey theory result is that $R(3,3) = 6$. That is, any red-blue coloring of a 6-clique has at least one monochromatic triangle in it.

Now, let $x = \log n$ be the size of the planted clique. Then there are $\binom{x}{6} = \Theta(x^6)$ 6-cliques, each of which must have a monochromatic triangle. Each triangle can be in at most $\binom{x}{3}$ 6-cliques, because that is the number of ways to choose the remaining three vertices. Thus, the number of monochromatic triangles is at least $\frac{\binom{x}{6}}{\binom{x}{3}} = \Theta(x^3)$. Since $x = \log n$, overall this is $\Theta(\log^3 n)$. □

We assume without loss of generality that Alice receives this number of triangles. Any node not in S is involved in $O(1)$ triangles before the clique planting, by our clustering-coefficient assumption. Using the maximum-degree assumption, simple probability, the uniform random selection of clique nodes, and the union bound, we now show that any node not in S has at most a constant number of edges into S whp. For ease of exposition, we use $\varepsilon = 1/2$ in our maximum-degree assumption. We could prove a similar theorem for any maximum degree of the form $O(n^{1-\varepsilon})$, for $\varepsilon > 0$.

Lemma 2. *With high probability, for any node $v \notin S$, the number of edges from v to S is $O(1)$*

Proof. Fix a node v and let X be a random variable giving the number of edges from v to S. By assumption, the maximum degree of any node in G is $\sqrt{n}$. So X is the sum of a set of at most $\sqrt{n}$

independent indicator random variables that are 1 with probability $(C\ln n)/n$. Thus

$$\begin{aligned} Pr(X \geq \lambda) &\leq \binom{\sqrt{n}}{\lambda}((C\ln n)/n)^{\lambda} \\ &\leq \left(\frac{\sqrt{n}e}{\lambda}\right)^{\lambda}\left(\frac{C\ln n}{n}\right)^{\lambda} \\ &\leq \left(\frac{\sqrt{n}e}{\lambda}\right)^{\lambda}((C\ln n)/n)^{\lambda} \\ &\leq \left(\frac{eC\ln n}{\sqrt{n}\lambda}\right)^{\lambda} \end{aligned}$$

Setting $\lambda = C_1$ for some constant C_1, we have:

$$\begin{aligned} Pr(X \geq \lambda) &\leq \left(\frac{eC\ln n}{\sqrt{n}\lambda}\right)^{\lambda} \\ &= \left(\frac{eC\ln n}{C_1\sqrt{n}}\right)^{C_1} \\ &= e^{(1+\ln(C/C_1)+\ln\ln n-1/2\ln n)C} \\ &\leq e^{-C_2\ln n} \end{aligned}$$

where the last line holds for any constant C_2, for C_1 and n sufficiently large.

Now let ξ be the event that for *any* node v that is not in S, v has at least C_1 neighbors in S. Then by a union bound,

$$\begin{aligned} Pr(\xi) &\leq ne^{-C_2\ln n} \\ &\leq e^{1-C_2\ln n} \\ &= n^{-C_3} \end{aligned}$$

where this holds for any constant C_3 for n and C_2 sufficiently large. □

Corollary 1. *With high probability, for any node $v \notin S$, the number of triangles that contain v and two nodes from S is $O(1)$*

Proof. By Lemma 2, with high probability, for any node $v \notin S$, v has $O(1)$ neighbors in S. This directly implies that v is in $O(1)$ triangles with 2 nodes in S. □

Any node $v \notin S$ has a constant number of triangles involving any node of S whp. Thus, since it started with $O(1)$ triangles before the clique planting, any node not in S is involved in $O(1)$ triangles whp.

We now argue Alice's subgraph $Q_a \subseteq S$ whp. The subgraph Q_a has triangle density $\Omega(\ln^2 n)$, since Alice received $\Theta(\ln^3 n)$ triangles of the clique with $\ln n$ nodes. In a subgraph of optimal triangle density ρ, any node participates in $\Omega(\rho)$ triangles. Otherwise, density would increase by dropping that node. Since any node $v \notin S$ is part of $O(1)$ triangles, it will not be in Q_a.

Since Q_a has triangle density $\Omega(\ln^2 n)$, and the maximum triangle density of a graph with x nodes is $O(x^3/x) = O(x^2)$, we have $|Q_a| = \Omega(\ln n)$. In fact, $|Q_a| = \Theta(\ln n)$ because $Q_a \subseteq S$.

The other nodes in S are neighbors of each node in Q_a. Therefore each such node will be adjacent to at least half the nodes in Q_a in G_a and/or G_b. Thus $S \subseteq Q_a \cup \bar{N}_a(Q_a) \cup \bar{N}_b(Q_a)$. If there are any stray nodes with high degree into Q_a (a low probability event), the clique-finding operation at the end will remove them. Because $|Q_a| = \Theta(\ln n)$, even exhaustive enumeration runs in polynomial time.

Chapter 3

Human and Automated Nodes in Social Networks

We found that social networks available in databases such as SNAP [15] do not appear to have a bounded number of triangles per node. That is, we found that publicly available social networks have more triangles on higher-degree vertices than predicted by [13]. We conjecture that social media networks contain human nodes and non-human nodes. For example, many Twitter accounts have automated behavior such as reciprocating all follower relationships [19], and businesses can buy fake followers for a penny each [12].

We document one attempt to remove non-human nodes based only on topology. These *s*-necessary triangles have properties required of vertices in large dense subgraphs such as cliques.

3.0.1 Definitions

For a vertex v, let ρ_v be the max x such that at least x neighbors w of v have $\rho_w \geq x$. This is an iterated notion motivated by the H-index concept used to rate publication productivity for researchers.

For a triangle t, $\rho_t = \min_{v \in t} \rho_v$

$P_v^s = |\{t : v \in t \text{ and } \rho_t \geq s\}|$

$V^s = \{v : P_v^s \geq s^2\}$

$T^s = \{t : \forall_{v \in T}, v \in V^s\}$ (these are the necessary triangles)

Since there is a constant-sized bound on the largest clique a human should be in, we tried removing s-necessary triangles (a condition for being in an s-node clique) for S around 500.

Aaron Kearns at the University of New Mexico created a simple classifier for human vs. non-human nodes in Twitter based on feature vectors. He developed an automated method for populating the feature vector from the Twitter page for a given account. He is currently training and validating the classifier against human judgement. Although initial studies for cleaning non-human elements by removing high-s-necessary nodes seemed promising, it does not appear to be a correct and stable solution.

The notion of s-necessary triangle is still relevant for finding dense subgraphs. There is a new kernel in the Mantevo [16] set of mini-applications, used to test new high-performance-computing systems, based on finding s-necessary triangles.

References

[1] Md. Faizul Bari, Raouf Boutaba, Rafael Esteves, Lisandro Zambenedetti Granville, Maxim Podlesny, Md Golam Rabbani, Qi Zhang, and Mohamed Faten Zhani. Data center virtualization: a survey. *IEEE Communications Surveys & Tutorials*, 15(2):909–927, 2013.

[2] Jonathan Berry, Michael Collins, Aaron Kearns, Phillips Cynthia A., Jared Saia, and Randy Smith. Cooperative computing for autonomous data centers. In *Proceedings of the 29th IEEE International Parallel and Distributed Processing Symposium*, May 2015.

[3] Justin Brickell and Vitaly Shmatikov. Privacy-preserving graph algorithms in the semi-honest model. In Bimal Roy, editor, *Advances in Cryptology - ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 236–252. Springer Berlin Heidelberg, 2005.

[4] T. Britton, M. Deijfen, and A. Martin-Löf. Generating simple random graphs with prescribed degree distribution. *Journal of Statistical Physics*, 124(6), September 2006.

[5] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 84–95, 2000.

[6] F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *PNAS*, 99:15879–15882, 2002.

[7] Easley David and Kleinberg Jon. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, New York, NY, USA, 2010.

[8] Yash Deshpande and Andrea Montanari. Finding hidden cliques of size $\sqrt{n/e}$ in nearly linear time. *arxiv*, 1304(7047v1), 2013.

[9] R.I.M. Dunbar. Social cognition on the internet: testing constraints on social network size. *Philosophical Transactions of the Royal Society B, Biological Sciences*, 367(1599):2192–2201, 2012.

[10] Cynthia Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[11] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[12] A. Horowitz and D. Horowitz. Watch for fakes on social media. *The Costco Connection*, page 17, May 2014.

[13] Tamara G. Kolda, Ali Pinar, Todd Plantenga, and C Seshadhri. A scalable generative graph model with community structure. *SIAM Journal on Scientific Computing*. to appear. Accepted March 2014. preprint available at `http://arxiv.org/abs/1302.6636`.

[14] Gueorgi Kossinets and Duncan J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, January 2006.

[15] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[16] Mantevo project. https://mantevo.org.

[17] Anne-Cecile Orgerie, Marcos Dias de Assuncao, and Laurent Lefevre. A survey on techniques for improving the energy efficiency of large scale distributed systems. *ACM Computing Surveys*, 46(4), 2014.

[18] Manoj M Prabhakaran and Amit Sahai. *Secure Multi-Party Computation*, volume 10. IOS press, 2013.

[19] Ryan A. Rossi, David F. Gleich, Assefaw H. Gebremedhin, and Md. Mostofa Ali Patwary. A fast parallel maximum clique algorithm for large sparse graphs and temporal strong components. *arxiv*, 1302(6256v1), 2013.

[20] Randy D. Smith, Jonathan Berry, and Cynthia A. Phillips. Ldrd final report: Geographically distributed graph algorithms. Technical Report SAND2013-0257, Sandia National Laboratories, Albuquerque, NM, January 2013.

DISTRIBUTION:

1	MS 0899	Technical Library, 9536 (electronic copy)
1	MS 0359	D. Chavez, LDRD Office, 1911

Sandia National Laboratories